Pedestrian Tracking Using Random Finite Sets

Stephan Reuter
Institute of Measurement, Control, and Microtechnology
University of Ulm
Ulm, Germany
Email: stephan.reuter@uni-ulm.de

Klaus Dietmayer
Institute of Measurement, Control, and Microtechnology
University of Ulm
Ulm, Germany
Email: klaus.dietmayer@uni-ulm.de

Abstract—For most multi-target tracking applications it is assumed that the movements of the objects are independent of each other. The validity of this assumption depends amongst others on the measurement rates of the sensors and the distance between the objects. In scenarios with a high object density the measurement rates for some of the objects may decrease due to short-time occlusions. Integrating the dependencies among the objects during occlusions should therefore improve the performance of the system. Within the finite set statistics (FISST) it is possible to model these dependencies and to integrate them into a Bayes filter. In this contribution a sequential Monte Carlo multi-target Bayes (SMC-MTB) filter based on FISST is used for pedestrian tracking. Furthermore, a model which avoids collisions of the pedestrians as well as a state dependent detection probability are integrated into the filter. The results of the SMC-MTB filter are evaluated using real sensor data and compared to the results of a CPHD filter.

Keywords: Tracking, Random Finite Sets, Filtering.

I. INTRODUCTION

In short-range tracking applications it is very likely that a large area of the sensors field of view is occluded due to extended objects which are located close to the sensors. In our application, two laser range scanners are mounted at two corners of a room. A possible scenario is shown in Fig. 1. We observe that the person at the top can not be observed by both sensors due to occlusions caused by two others.

In most applications a detection probability is used in tracking algorithms to model missed detections (false negatives), i.e. the object is visible to the sensor, but no measurement is received [1], [2]. In our scenario, the detection probability explicitly depends on the state of the target in consequence of the possible occlusions caused by other extended objects. As shown in [3] the state uncertainty of an object depends on the shape of the occluded area. Thus, a particle filter will be used to track occluded objects because it is capable to represent all types of distributions and it is not limited to Gaussian distributions.

Moreover, most multi-target tracking algorithms are based on the assumption that the motion of all objects are statistically independent. Considering extended objects which are very close to each other, it is obvious, that this assumption is not fulfilled any more since the objects can not be located at the same place. In [4], [5] the integration of environmental constraints, like e.g. "ships can only be located on the sea", into particle filters has been investigated. The multi-target Bayes filter proposed by Mahler [6] offers the opportunity to model directly the dependency between objects. Since the multi-target Bayes filter propagates the multi-target posterior density recursively in time, several set-integrals have to be evaluated which is computationally intractable. Thus, approximations of the multi-target Bayes filter using Sequential Monte Carlo (SMC) methods have been proposed [6]–[10]. For a large number of targets these approximations are still computationally demanding. Other approximations like the Probability Hypothesis Density (PHD) or the Cardinalized PHD (CPHD) filter [11], [12] require less computing time at the cost of losing the possibility to model interactions between the objects.

In this contribution, a Sequential Monte Carlo (SMC) approximation of the multi-target Bayes filter is applied to pedestrian tracking in occlusion scenarios. Further, a graph based calculation method of the multi-target likelihood function is proposed. In addition to that, a grid map based approach to estimate the state dependent detection probability is introduced and integrated into the filter. Since the multi-target Bayes filter...
allows to model dependencies between objects, a anti-collision function is developed to check the validity of the prediction step. In order to show the potential of the multi-target Bayes filter in occlusion scenarios, the results are compared to those of a CPHD filter.

The contribution is organized as follows: First, the single-target and multi-target Bayes filter are introduced and shortly reviewed. In section III the SMC implementation of the multi-target particle filter is presented. Then, an approximative method to calculate the state dependent detection probability is developed. Finally, in section V the tracking results of the SMC multi-target Bayes filter are evaluated using real-world sensor data and compared to the results of a CPHD filter.

II. BAYESIAN TRACKING

In the Bayesian tracking approach, a posterior probability density function (pdf) is used to represent all information about the state of an object. Further, an optimal estimate of the state and a measure of accuracy may be obtained from this pdf. Since each new measurement increases the information about the state of the object, the pdf has to be updated with every received measurement. Thus, a recursive filtering algorithm which processes the data sequentially is more convenient than batch processing since it is not necessary to store already processed measurements and reprocess the existing data if new measurements are received. The tracking filter consists of a predictor and a corrector step. In the predictor step a motion model is used to calculate an a priori pdf for the next measurement time. In the corrector step the received measurement is used to update the a priori pdf. In order to show the analogy between the single-target and the multi-target Bayes (MTB) filter [6], we first review the single-target Bayes filter.

A. Single-Target Bayes Filter

In the predictor step of the single-target Bayes filter, the predicted pdf for time \( k + 1 \) is obtained via the Chapman-Kolmogorov equation:

\[
f_{k+1|k}(x|Z^k) = \int f_{k+1|k}(x'|x) \cdot f_{k|k}(x|Z^k) dx', \tag{1}
\]

where \( Z^k : z_1, ..., z_k \) is the sequence of all measurements up to time step \( k \). The probability density function \( f_{k+1|k}(x'|x) \) describes the motion model of the target under the assumption that the motion is independent on the previous measurements.

In the corrector step, the received measurement \( z_{k+1} \) is used to calculate the posterior pdf

\[
f_{k+1|k+1}(x|Z^{k+1}) = \frac{f_{k+1|k+1}(z_{k+1}|x) \cdot f_{k+1|k+1}(x|Z^k)}{f_{k+1}(Z^{k+1})}, \tag{2}
\]

where \( f_{k+1}(Z^{k+1}) \) is the Bayes normalization factor.

Since equations (1) and (2) of the Bayes filter can not be solved analytically, the filter is either approximated using SMC methods [13] or by the Kalman filter and its extensions [1].

B. Multi-Target Bayes Filter

The multi-target Bayes filter is an extension of the single-target Bayes filter. Instead of a vector \( x \) the multi-target Bayes filter [6] uses a random finite set \( X \).

Analog to (1), the predictor step is given by

\[
f_{k+1|k}(X|Z^{k+1}) = \int f_{k+1|k}(X'|X) \cdot f_{k|k}(X|Z^k) dX', \tag{4}
\]

where a set integral [6] is used instead of a vector integral.

The multi-target Markov model \( f_{k+1|k}(X|X') \) used in the predictor step is more complex than the single-target motion model \( f_{k+1|k}(x|x') \), because it has to represent the motion of the targets as well as target appearance and disappearance. Assuming the case that each target persists with a survival probability \( p_S(x_i) \) and no new targets appear, the probability that there are no targets at all in the scene at time \( k + 1 \) is

\[
f_{k+1|k}(\emptyset|X') = \prod_{i=1}^{n'} (1 - p_S(x_i)). \tag{5}
\]

The multi-target Markov model is finally given by

\[
f_{k+1|k}(X|X') = f_{k+1|k}(\emptyset|X') \sum_{\theta} \prod_{i=1}^{n'} \frac{p_S(x_i) \cdot f_{k+1|k}(x_i|x_i')}{1 - p_S(x_i)} \tag{6}
\]

where the summation is calculated for all associations \( \theta : \{1, ..., n'\} \to \{0, 1, ..., m\} \).

The corrector step of the multi-target Bayes filter is also analog to (2) and (3):

\[
f_{k+1|k+1}(X|Z^{k+1}) = \frac{f_{k+1|k}(Z_{k+1}|X) \cdot f_{k+1|k}(X|Z^k)}{f_{k+1}(Z^{k+1})}, \tag{7}
\]

with the Bayes normalization factor

\[
f_{k+1}(Z^{k+1}) = \int f_{k+1}(Z_{k+1}|X) \cdot f_{k+1|k}(X|Z^k) dX. \tag{8}
\]

As in the predictor step a set integral is used instead of a vector integral.

In order to calculate the corrector step, the multi-target likelihood function \( f_{k+1}(Z_{k+1}|X) \) is needed. The multi-target likelihood function averages over all association hypotheses \( \theta : \{1, ..., n\} \to \{0, 1, ..., m\} \) and takes them into account equally since there is no a priori knowledge available. In case of missed detections and no false alarms, the likelihood function is given by

\[
f_{k+1}(Z_{k+1}|X) = \sum_{\theta \in \Theta_{(k+1)}} \prod_{i=1}^{n} \frac{p_D(x_i) \cdot f_{k+1|k}(z_{\theta(i)}|x_i) \cdot f_{k+1|k}(x_i)}{1 - p_D(x_i)} \tag{9}
\]

where each association hypothesis is represented by one summand and \( p_D(x_i) \) is the detection probability. The factor \( f_{k+1}(\emptyset|X) \) is given by

\[
f_{k+1}(\emptyset|X) = \prod_{i=1}^{n} (1 - p_D(x_i)). \tag{10}
\]

Analog to the single-target case, it is not possible to calculate (4) and (7) analytically, but the MTB filter can also be approximated using SMC particle implementations. Further, the
MTB filter can be approximated by the probability hypothesis density (PHD) filter and the cardinalized PHD (CPHD) filter [6]. Both filters propagate the first moment of the multi-target posterior probability density function. Since the PHD filter tends to be unstable in the estimated number of targets, the CPHD filter additionally propagates the cardinality distribution. The PHD and CPHD filters can be implemented using SMC methods or Gaussian Mixtures [6], [8], [14], [15].

III. SMC MULTI-TARGET BAYES FILTER

In a single-target particle filter the posterior distribution $f_{x_k}(x|Z^k)$ is approximated by a collection of state vectors $x_{k|j}^1, \ldots, x_{k|j}^v$ and the according positive weights $w_{k|j}^1, \ldots, w_{k|j}^v$ with $\sum_{i=1}^v w_{k|j}^i = 1$, which are propagated through time.

Correspondingly, the multi-target posterior distribution $f_{x_k}(X|Z^{k+1})$ can be approximated by a collection of state sets $X_{k|j}^1, \ldots, X_{k|j}^v$ and positive weights $w_{k|j}^1, \ldots, w_{k|j}^v$ with $\sum_{i=1}^v w_{k|j}^i = 1$.

In [6]–[8] SMC implementations of the multi-target Bayes filter have been proposed. The notations used in this section are based on the equations in [6].

In the multi-target Bayes filter, the predictor step is more complex than in the single-target case. Here, the prediction step of the multi-target particle filter consists of two parts: persistence and appearance. Depending on the application it might be necessary to model spawning, too. We assume a constant persistence probability of $p_S = 0.99$ for each particle $j$ of a finite set $X_{k|j}^i$.

The number of persisting targets $n$ and the indexes of the persisting particles of the set $X_{k|j}^i = \{x_{1|j}^i, \ldots, x_{n|j}^i\}$ are determined by drawing a sample of a multi-target multi-Bernoulli distribution [6]. Under the assumption that target motions are independent, for each of the persisting particles $j = j_1, \ldots, j_{n|\text{persist}}$ we draw the predicted particles according to the motion model $f_{x_{k+1}|x_j|j}(x_{k+1}|x_j|j)$.

In order to integrate the dependencies between the motions of the pedestrians, an approximation of the social force model [16] is used. In the social force model, force vectors are used to change the moving direction of a pedestrian depending on the environment. All possible moving directions of the pedestrian, neglecting the influence of other pedestrians, are already included in the prediction step of the particle filter. Thus, we propose to use the social force model as an additional weighting factor for the predicted multi-target particles instead of using the model to modify the directions.

We assume, that the shape of a pedestrian is approximately circular with a radius of $r_p = 0.2m$. The actual distance $d_{rs}$ between two pedestrians $r$ and $s$ is used to calculate the likelihood function

$$l_{rs} = \begin{cases} 0 & \text{if } d_{rs} < 2r_p \\ 1 - \exp\left(-\frac{d_{rs} - 2r_p}{2\sigma_f}\right) & \text{otherwise} \end{cases} \tag{12}$$

where $\sigma_f$ is used to model the preferred distance to other pedestrians. In our application, we use $\sigma_f = 0.1m$. Since the likelihood function ensures a minimal distance between the pedestrians and avoids collisions of the pedestrians, it is called anti collision function in the following. The anti collision function has to be evaluated for all possible pairs of $r$ and $s$. Finally, the adapted weights of the multi-target particles are given by

$$w_{k+1|j}^{SF} = \min\left(l_{rs}\right) \cdot w_{k+1|j}^i \tag{13}$$

where $w_{k+1|j}^i$ is the weight of the predicted multi-target particle. If the distance between any of the pedestrians in the set is smaller than $2r_p$, the weight of this set is set to zero. In contrast, $w_{k+1|j}^{SF} \approx w_{k+1|j}^i$ if the minimum distance is greater than $2r_p + 0.1m$.

The appearing targets are created using a measurement driven appearance model, i.e. each measurement of the previous time step that could not be associated with any of the particles, creates a new particle in the set with probability $p_B$.

Finally, the predicted particle set is given by the union of these two sets:

$$X_{k+1|j}^i = X_{k+1|j|\text{persist}}^i \cup X_{k+1|j|\text{appear}}^i \tag{14}$$

Due to the use of an appearance model, the filter can be initialized using a null-target prior, i.e. there are no targets in the scene.

In the corrector step the multi-target likelihood $f_{k+1}(Z_{k+1}|X_{k+1|j}^i)$ has to be calculated for all particle sets $X_{k+1|j}^i$. Here, the assumptions that an object generates no more than one measurement and that a measurement is generated by no more than one target are used. First, the spatial likelihoods $f_{x_{k+1}}^i(x_{k+1}|x_j)$ are pre-calculated for all possible associations within the current set. Then, similar to the implementation of the Joint Integrated Probabilistic Data Association (JIPDA) [17] filter in [2] we propose to use a tree structure to calculate the multi-target likelihood. In Fig. 2 the tree for a particle set of dimension two with two measurements is shown. Each branch of the tree represents one factor of the product in equation (9). The product of all branches from the root of the tree to a leaf equals one summand of (9).

Figure 2. Tree structure to evaluate the multi-target likelihood function: first value of each node represents the index of the particle, the second value represents the number of the measurement or the missed detection (0) [2].

The corrector step is the computationally intensive part of the algorithm, since the association tree has to be evaluated for each of the multi-target particles. Since the depth $n$ of the association tree is given by the number of objects in
the set and the branching factor is given by the number of the measurements \( m \) and the additional symbol for a missed detection, an approximate upper bound for the time complexity of the tree is given by

\[
O(m^{m+1}).
\]

Due to the exponential increase of the complexity, the evaluation of the tree is only possible for small numbers of measurements and objects. In order to reduce the complexity, gating can be integrated into the calculation of the tree by truncating the calculation of branches with a likelihood close to zero. In case of 10 measurements and 6 objects the evaluation time for 2000 trees is approximately 300 ms using a truncation threshold of 0.1. Finally, the weights of the particle sets are normalized by

\[
w_{k+1}^{i} = \frac{f_{k+1} \left( Z_{k+1} | X_{k+1}^{i} \right) \cdot w_{k}^{i}}{\sum_{i=1}^{N} f_{k+1} \left( Z_{k+1} | X_{k+1}^{i} \right) \cdot w_{k}^{i}},
\]

and the a posteriori particle sets are given by

\[
X_{k+1}^{i} = \sum_{i=1}^{N} w_{k+1}^{i} X_{k+1}^{i} | X_{k+1} = X_{k+1}^{i}.
\]

As in the single-target case, the SMC-MTB filter tends to degenerate. Thus, a systematic resampling algorithm is used.

The number of targets \( \hat{n} \) is estimated by

\[
\hat{n}_{k+1} \triangleq \frac{1}{\nu} \sum_{i=1}^{N} w_{k+1}^{i} | X_{k+1}^{i} \triangleq \frac{1}{\nu} \sum_{i=1}^{N} w_{k+1}^{i} X_{k+1}^{i},
\]

where \( | X_{k+1} \triangleq \frac{1}{\nu} \sum_{i=1}^{N} w_{k+1}^{i} X_{k+1}^{i} \) is the number of states of the finite set. Since the measurement noise is modeled by a Gaussian distribution, the posterior pdf of all visible objects is approximately Gaussian. Thus, the EM-clustering algorithm [18] is used to estimate the states of the objects. The EM-algorithm uses all particles of the sets with dimension \( \hat{n} \) to calculate the \( \hat{n} \) Gaussian distributions with according weights. The EM-algorithm is initialized with the states of one of the particle sets with dimension \( \hat{n} \) in order to achieve robust estimation results.

IV. DETECTION PROBABILITY

In our scenario, the detection probability depends on the actual state of the objects due to the fact that they are often occluded due to other objects. We propose to use the principles of the occupancy grid mapping approach [19] to calculate the state dependent detection probability. The main idea of the occupancy grid mapping is to divide the environment into grid cells and assign every grid cell \( m \) an occupancy likelihood \( p(m | z_{t}, g) \) that is based on all measurements up to time \( t \). An occupancy likelihood \( p(m) = 0 \) denotes a cell which is not occupied by an object while \( p(m) = 1 \) denotes an occupied cell. For every new measurement, two steps have to be executed: building a measurement grid and updating the occupancy grid.

The measurement grid basically divides the grid map into three regions: free, occupied, and hidden cells. First, each cell \( m \) of the measurement grid \( g \) is initialized with \( g(m) = 0.5 \) because no information about the occupancy of the cell is available. All distance measurements of a laser scan are transformed to the grid coordinates and the likelihood of the corresponding cells \( m \) are set to \( g(m) = 1 \). Since a grid cell resolution of 5cm is used, the size of the pedestrians has to be incorporated into the construction of the measurement grid. An approximate solution is to denote all cells which are less than 50cm behind a measurement as occupied. Further, all cells between the sensor and the measurements are denoted as free space.

In order to determine the detection probability, the information of the two measurement grids has to be combined. Since it is not necessary to detect the objects with both sensors to generate a measurement for the tracking algorithm, a grid cell is only marked as occluded if it is occluded for both sensors. Thus, it is sufficient to calculate the binary detection map

\[
d(m) = \begin{cases} 
0 & \text{if } g_{1}(m) = g_{2}(m) = 0.5, \\
1 & \text{otherwise}, 
\end{cases}
\]

where \( g_{1} \) and \( g_{2} \) are the measurement grids of the two sensors.

In Fig. 3 the binary detection map is illustrated using magenta circles to represent the occluded grid cells. The scene contains an occluded person located at \((x, y) \approx (-0.9m, 5.5m)\). We observe, that the area around the center of the occluded person is marked as occluded, while the areas directly behind the measurements of the visible persons are not marked as occluded although they may not be observed by any of the sensors.

Next, the size of the objects has to be integrated into the detection map in order to provide correct results at the border separating detectable and occluded areas. Assuming that all pedestrians are of approximately same size, the size can be integrated by convolving the detection map with a circular kernel function [3]. Fig. 4 shows a grid map representing the spatial detection probability for the situation of Fig. 3. The convolution with the kernel function removes the edges and smooths the map.

V. RESULTS

The SMC-MTB filter was applied to real sensor data of a scenario with up to seven persons. Since the measurement set of one of our laser range finders consists of around 800 points, using each of the points as a measurement in the SMC-MTB filter would result in a very high computational load. Thus, we cluster the points into segments. In order to reduce the loss of information between the raw measurements and the segmented data, we use the fuzzy segmentation method introduced in [20] which generates several segmentation hypotheses with associated confidence values instead of one hard decision. Further, measurements of the background (e.g. walls) are removed from the measurement set using the occupancy grid mapping approach.

In Fig. 5 the results of the segmentation algorithm are shown. The algorithm combines the results of both scanners by fitting an ellipse using the least squares approach introduced in
Figure 3. Bird’s eye view of the measurements of our laser scanners (black, blue and green stars) overlaid with the binary detection map (magenta circles). Black stars correspond to non-moving obstacles, green and blue stars correspond to moving objects.

[21] into the measurement points. We observe, that each of the five visible persons generates at least one segmentation result. Additionally, there is an occluded person in the scene which is located at \((x, y) \approx (-0.9m, 5.5m)\). Since the implemented SMC-MTB filter does not use the size of the objects, the centers of the ellipses are used as measurements.

The SMC-MTB filter uses a constant velocity motion model. Further, the state dependent detection probability and the anti-collision function presented in the previous sections are used. The laser range finders deliver measurements at a rate of 12.5Hz. Thus, one time step \(k\) corresponds to 0.08 seconds. The investigated sequence is split into five parts:

a) Initialization
b) Six persons, none of the persons is occluded for more than four measurements in a row
c) One of the persons is completely occluded during this part (36 measurements)
d) Occluded person re-appears, one of the other persons is occluded for eleven measurements
e) An additional person enters the scene.

A. SMC-Multi-Target Bayes Filter

In Fig. 6 the mean value of the estimated number of targets \(\hat{n}\) and the standard deviation \(\sigma\) determined by a SMC-MTB filter with 10000 particle sets are shown. Part a) shows the initialization procedure. After about 1.5 seconds the estimated number of persons is equal to the true number of persons. In parts b) and d) the estimated number of persons is very accurate. If we receive measurements of all persons and no false alarms, the standard deviation is very small. On the other hand, the standard deviation increases immediately in case of missed detections or false alarms. In part c) the estimated number of persons decreases slowly due to the missing measurements of one of the persons and the survival
probability $p_S = 0.99$. As expected, the standard deviation increases during the occlusion. Since there is still a large amount of particle sets that represent the state with 6 persons, the estimated number of persons increases rapidly to 6 as soon as the person leaves the occlusion in part $d$). In part $e)$, the estimated number of persons increases within 0.5 seconds to the new number of persons.

In Fig. 7 the particles and the results of the EM algorithm at $k = 300$ are shown. Although the person at $(-0.9 \text{m}, 5.5 \text{m})$ is already occluded for more than one second, the particles of the six persons are still well separated and the EM algorithm successfully fits the six Gaussian distributions. The excellent performance is due to the use of the anti collision function (which can be applied because of the particle sets) and the state dependent detection probability.

Next, the dependence of the tracking performance on the number of multi-target particles is evaluated. The results of a filter using 2000 multi-target particles are compared to the results of a filter using 10000 multi-target particles.

In Fig. 8 the mean value for the estimated number of objects for 10 Monte Carlo runs is shown. Due to the small birth probability of $p_B = 0.001$, the filter with more multi-target particles needs less time to reach the true number of objects (parts $a$ and $b$)). If none of the objects is occluded for more than 0.3 seconds, the performance of both filters is similar (part $b$). Even during the long-term occlusion in part $c$) there is no significant difference. But the filter with 10000 multi-target particles outperforms the one with 2000 in part $d$) when the object re-appears and its estimated number of objects is more stable in part $e$).

Fig. 9 shows the standard deviation of the number of objects. After the filters finished the initialization in all runs ($k \approx 75$), the standard deviation of both filters is similar in part $b$). The long-term occlusion in part $c$) leads to a higher standard deviation of the filter with $N = 2000$ in part $d$) and $e$) since there are some more occlusions in these parts such that the filter does not successfully tune to the true number of objects in all runs.
We observe that the necessary number of particles depends on the actual situation. In order to be able to handle occlusion scenarios reliably, a much higher number of particles is necessary, since the uncertainty about a person’s state can be very high. On the other hand, situations without occlusions can be successfully handled with less particles without influence on the performance.

B. CPHD vs. Multi-Target Bayes Filter

In Fig. 10 the tracking performance of the SMC-MTB filter is compared to the results of a CPHD particle filter. The CPHD uses the same detection map as the SMC-MTB filter. Since the SMC-MTB filter uses 10000 multi-target particles which corresponds to 60000 single-target particles if each of the finite sets consists of 6 state vectors, the CPHD filter uses 60000 single-target particles. The estimated number of persons of the CPHD filter are close to those of the SMC-MTB filter if all persons are visible to the sensor or if they are only occluded for a short time. In case of a long-term occlusion (part c) the SMC-MTB filter outperforms the CPHD filter.

In Fig. 11 the distribution of the particles of the CPHD filter and the results of the EM algorithm at \( k = 300 \) are shown. Since the particles of the CPHD filter are not represented as sets, we use a PHD-based approach [7] to initialize the EM-algorithm in order to improve the clustering results. Comparing the results of the CPHD filter to the results shown in Fig. 7 we observe, that the blue and cyan clusters are not as well separated. This effect is caused by the fact that the particles are not represented by sets in the CPHD filter and consequently the anti collision function can not be applied in the prediction step in the same manner. Alternatively, the estimates of the CPHD filter at time \( k \) could be used to adapt the distribution which is used to predict the particles.

VI. CONCLUSIONS

In this contribution a multi-target particle filter using random finite sets has been successfully applied to pedestrian tracking with laser range scanners. In order to simplify the calculation of the multi-target likelihood, a graph based calculation method has been developed. Further, a method to calculate a state dependent detection probability based on a grid map has been introduced. Especially in occlusion...
scenarios the random finite set approach delivers excellent results and outperforms the CPHD approximations.

In future we plan to integrate more interactions between the pedestrians and the environment by applying further parts of the social force model. Further, an implementation of the filter using a graphics processing unit (GPU) to achieve real-time performance is considered. In order to be able to compare the accuracy of the estimated states of the filters during occlusions, additional sensors will be integrated into the setup to achieve reference measurements for the states of the objects.

VII. ACKNOWLEDGMENTS

This work is done within the Transregional Collaborative Research Center SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

REFERENCES